

# Cooperative Multiagent Learning

Enric Plaza and Santiago Ontaño

IIIA – Artificial Intelligence Research Institute  
CSIC – Spanish Council for Scientific Research  
Campus UAB, 08193 Bellaterra, Catalonia (Spain)  
Vox: +34-93-5809570, Fax: +34-93-5809661  
`{enric,santi}@iiia.csic.es`  
<http://www.iiia.csic.es>

**Abstract.** Cooperation and learning are two ways in which an agent can improve its performance. Cooperative Multiagent Learning is a framework to analyze the tradeoff between cooperation and learning in multiagent systems. We focus on multiagent systems where individual agents are capable of solving problems and learning using CBR (Case-based Reasoning). We present several collaboration strategies for agents that learn and their empirical results in several experiments. Finally we analyze the collaboration strategies and their results along several dimensions, like number of agents, redundancy, CBR technique used, and individual decision policies.

## 1 Introduction

Multiagent systems offer a new paradigm to organize AI applications. Our goal is to develop techniques to integrate lazy learning into applications that are developed as multiagent systems. Learning is a capability that together with autonomy is always defined as a feature needed for full-fledged agents. Lazy learning offers the multiagent systems paradigm the capability of autonomously learning from experience. In this paper we present a framework for collaboration among agents that use Case-based Reasoning (CBR) and some experiments illustrating the framework.

A distributed approach for lazy learning in agents that use CBR (case-based reasoning) makes sense in different scenarios. Our purpose in this paper is to present a multiagent system approach for distributed case bases that can support these different scenarios. A first scenario is one where cases themselves are owned by different partners or organizations. These organizations can consider their cases as assets and they may not be willing to give them to a centralized “case repository” where CBR can be used. In our approach each organization keeps their private cases while providing a CBR agent that works with them. Moreover, the agents can collaborate with other agents if they keep the case privacy intact and they can improve their performance by cooperating. Another scenario involves scalability: it might be impractical to have a centralized case base when the data is too big.

Our research focuses on the scenario of separate case bases that we want to use in a decentralized fashion by means of a multiagent system, that is to say a collection of CBR agents that manage individual case bases and can communicate (and collaborate) with other CBR agents. From the point of view of Machine Learning (ML) our approach can be seen as researching the issues of learning with distributed or “partitioned” data: how to learn when each learning agent is able to see only a part of the examples from which to learn. This approach is related to the work in ML on *ensembles* or committees of classifiers (we explain this relationship later in § 6). The main difference is that ensembles work on collection of classifiers that see all data but treat them differently, while our focus a collection of agents each having a view of part of the data (that in the extreme case can be completely exclusive). In this paper we show several strategies for collaboration among learning agents and later we analyze their results in terms of ML concepts like the error in terms of bias plus variance and the “ensemble effect”.

Form the point of view of agent systems, we focus on multiagent systems and not on distributed applications. In distributed applications there are some overall goals that govern the different parts performing distributed processing, and their coordination is decided at design time, it is not decided by the constituent parts. In a multiagent system, agents have autonomy—i.e. they have individual goals that determine when it is in their interest to collaborate with others, and when not. In our approach, the agents have autonomy given by individual data (the cases from which they learn) and individual goals (solving problems and improving their performance), and they only collaborate when it can further their goals.

## 2 Collaboration Strategies

A collaboration strategy in a *MAC* system establishes a coordination structure among agents where each agent exercises individual choice while achieving an overall effect that is positive both for the individual members and the whole system. Specifically, a collaboration strategy involves two parts: interaction protocols and decision policies. The interaction protocols specify the admissible pattern of message interchange among agents; e. g. a simple protocol is as follows: agent *A* can send a *Request* message to agent *B* and then agent *B* can reply with an *Accept* message or a *Reject* message. Interaction protocols specify *interaction states* whose meaning is shared by the agents; in our example, agent *A* knows that it’s up to agent *B* to accept or not the request, and agent *B* knows that agent *A* is expecting an answer (usually in a time frame specified in the message as an expiration time). An interaction state then requires some agent to make a decision and act accordingly: the *decision policies* are the internal, individual procedures that agents use to take those decisions following individual goals and interests.

In the following sections we will show several strategies for collaboration in the framework of interaction protocols for committees of agents. Since interaction

protocols for committees are quite similar we will focus on different individual decision policies that can be used while working in committees.

## 2.1 Multiagent CBR

A multiagent CBR ( $\mathcal{MAC}$ ) system  $\mathcal{M} = \{(A_i, C_i)\}_{i=1\dots n}$  is composed on  $n$  agents, where each agent  $A_i$  has a case base  $C_i$ . In this framework we restrict ourselves to analytical tasks, i.e. tasks (like classification) where the solution is achieved by selecting from an enumerated set of solutions  $K = \{S_1 \dots S_K\}$ .

When an agent  $A_i$  asks another agent  $A_j$  help to solve a problem the interaction protocol is as follows. First,  $A_i$  sends a problem description  $P$  to  $A_j$ . Second, after  $A_j$  has tried to solve  $P$  using its case base  $C_j$ , it sends back a message that is either `:sorry` (if it cannot solve  $P$ ) or a solution endorsement record (SER). A SER has the form  $\langle \{(S_k, E_k^j)\}, P, A_j \rangle$ , where the collection of *endorsing pairs*  $(S_k, E_k^j)$  mean that the agent  $A_j$  has found  $E_k^j$  cases in case base  $C_j$  endorsing solution  $S_k$ —i.e. there are a number  $E_k^j$  of cases that are relevant (similar) for endorsing  $S_k$  as a solution for  $P$ . Each agent  $A_j$  is free to send one or more endorsing pairs in a SER record.

## 2.2 Voting Scheme

The voting scheme defines the mechanism by which an agent reaches an aggregate solution from a collection of SERs coming from other agents. The principle behind the voting scheme is that the agents vote for solution classes depending on the number of cases they found endorsing those classes. However, we do not want that agents having a larger number of endorsing cases may have an unbounded number of votes regardless of the votes of the other agents. Thus, we will define a normalization function so that each agent has one vote that can be for a unique solution class or fractionally assigned to a number of classes depending on the number of endorsing cases.

Formally, let  $\mathcal{A}^t$  the set of agents that have submitted their SERs to agent  $A_i$  for problem  $P$ . We will consider that  $A_i \in \mathcal{A}^t$  and the result of  $A_i$  trying to solve  $P$  is also reified as a SER. The vote of an agent  $A_j \in \mathcal{A}^t$  for class  $S_k$  is

$$Vote(S_k, A_j) = \frac{E_k^j}{c + \sum_{r=1\dots K} E_r^j}$$

where  $c$  is a constant that on our experiments is set to 1. It is easy to see that an agent can cast a fractional vote that is always less than 1. Aggregating the votes from different agents for a class  $S_k$  we have ballot

$$Ballot^t(S_k, \mathcal{A}^t) = \sum_{A_j \in \mathcal{A}^t} Vote(S_k, A_j)$$

and therefore the winning solution class is

$$Sol^t(P, \mathcal{A}^t) = \arg \max_{k=1\dots K} Ballot(S_k, \mathcal{A}^t)$$

i.e., the class with more votes in total. We will show now two collaboration policies that use this voting scheme.

### 3 Committee Policy

In this collaboration policy the member agents of a  $\mathcal{MAC}$  system  $\mathcal{M}$  are viewed as a committee. An agent  $A_i$  that has to solve a problem  $P$ , sends it to all the other agents in  $\mathcal{M}$ . Each agent  $A_j$  that has received  $P$  sends a solution endorsement record  $\langle \{(S_k, E_k^j)\}, P, A_j \rangle$  to  $A_i$ . The initiating agent  $A_i$  uses the voting scheme above upon all SERs, i.e. its own SER and the SERs of all the other agents in the multiagent system. The final solution is the class with maximum number of votes.

The next policy, *Bounded Counsel*, is based on the notion that an agent  $A_i$  tries to solve a problem  $P$  by himself and if  $A_i$  “fails” to find “good” solution then  $A_i$  asks counsel to other agents in the  $\mathcal{MAC}$  system  $\mathcal{M}$ .

Let  $E_P^i = \{(S_k, E_k^i)\}$  the endorsement pairs the agent  $A_i$  computes to solve problem  $P$ . For an agent  $A_i$  to decide when it “fails” we require that each agent in  $\mathcal{M}$  has a predicate *Self-competent*( $P, E_P^i$ ). This predicate determines whether or not the solutions endorsed in  $E_P^i$  allow the agent to conclude that there is a good enough solution for  $P$ .

#### 3.1 Bounded Counsel Policy

In this policy the agents member of a  $\mathcal{MAC}$  system  $\mathcal{M}$  try first to solve the problems they receive by themselves. Thus, if agent  $A_i$  receives a problem  $P$  and finds a solution that is satisfactory according to the *termination check* predicate, the solution found is the final solution. However, when an agent  $A_i$  assesses that its own solution is not reliable, the Bounded Counsel Policy tries to minimize the number of questions asked to other agents in  $\mathcal{M}$ . Specifically, agent  $A_i$  asks counsel only to one agent, say agent  $A_j$ . When the answer of  $A_j$  arrives the agent  $A_i$  uses the termination check. If the termination check is true the result of the voting scheme at that time is the final result, otherwise  $A_i$  asks counsel to another agent—if there is one left to ask, if not the process terminates and the voting scheme determines the global solution.

The termination check works, at any point in time  $t$  of the Bounded Counsel Policy process, upon the collection of solution endorsement records (SER) received by the initiating agent  $A_i$  at time  $t$ . Using the same voting scheme as before, Agent  $A_i$  has at any point in time  $t$  a plausible solution given by the winner class of the votes cast so far. Let  $V_{max}^t$  be the votes cast for the current plausible solution,  $V_{max}^t = \text{Ballot}^t(\text{Sol}^t(P, \mathcal{A}^t), \mathcal{A}^t)$ , the termination check is a boolean function  $\text{TermCheck}(V_{max}^t, \mathcal{A}^t)$  that determines whether there is enough difference between the majority votes and the rest to stop and obtain a final solution. In the experiments reported here the termination check function is the following

**Table 1.** Average precision and standard deviation for a case base of 280 sponges pertaining to three classes. All the results are obtained using a 10-fold cross validation.

Policy	3 Agents		4 Agents		5 Agents		6 Agents		7 Agents	
	$\mu$	$\sigma$	$\mu$	$\sigma$	$\mu$	$\sigma$	$\mu$	$\sigma$	$\mu$	$\sigma$
<i>Isolated</i>	83.2	6.7	82.5	6.4	79.4	8.4	77.9	7.6	75.8	6.8
<i>Bounded</i>	87.2	6.1	86.7	6.5	85.1	6.3	85.0	7.3	84.1	7.0
<i>Committee</i>	88.4	6.0	88.3	5.7	88.4	5.4	88.1	6.0	87.9	5.9

$$TermCheck(V_{max}^t, \mathcal{A}^t) = \frac{V_{max}^t}{Max(1, Ballot(S_k, \mathcal{A}^t) - V_{max}^t)} \geq \eta$$

i.e. it checks whether the majority vote  $V_{max}^t$  is  $\eta$  times bigger than the rest of the ballots. After termination the global solution is the class with maximum number of votes at that time.

### 3.2 Experimental Setting

In order to compare the performance of these policies, we have designed an experimental suite with a case base of 280 marine sponges pertaining to three different orders of the *Demospongiae* class (*Astrophorida*, *Hadromerida* and *Axinellida*). The goal of the agents is to identify the correct biological order given the description of a new sponge.

We have experimented with 3, 4, 5, 6 and 7 agents using LID [1] as the CBR method. The results presented here are the result of the average of 5 10-fold cross validation runs. Therefore, as we have 280 sponges in our case base, in each run 252 sponges will form the training set and 28 will form the test set.

In an experimental run, training cases are randomly distributed to the agents (without repetitions, i.e. each case will belong to only one agent case base). Thus, if we have  $n$  agents and  $m$  examples in the training set, each agent should have about  $m/n$  examples in its case base. Therefore increasing the number of agents in our experiments their case-base size decreases. When all the examples in the training set have been distributed, the test phase starts.

In the test phase, for each problem  $P$  in the test set, we randomly choose an agent  $A_i$  and send  $P$  to  $A_i$ . Thus, every agent will only solve a subset of the whole test set. If testing the isolated agents scenario,  $A_i$  will solve the problem by itself without help of the other agents. And if testing any of the collaboration policies,  $A_i$  will send  $P$  to some other agents.

We can see (Table 1) that in all the cases we obtain some gain in accuracy compared to the isolated agents scenario. The Committee policy is always better than the others; however this precision has a higher cost since a problem is always solved by every agent. If we look at *Bounded Counsel* policy we can see it is much better than the isolated agents, and slightly worse than the Committee policy—but it is a cheaper policy since less agents are involved.

A small detriment of the system’s performance is observable when we increase the number of agents. This is due to the fact that the agents have a more

reduced number of training cases in their case bases. A smaller case base has the effect of obtaining less reliable individual solutions. However, the global effect of reducing accuracy appears on *Bounded Counsel* but not on the Committee policy. Thus, the Committee policy is quite robust to the effect of diminishing reliability individual solutions due to smaller case bases. This result is reasonable since the Committee policy always uses the information available from all agents. A more detailed analysis can be found in [10]

The *Bounded Counsel* policy then only makes sense if we have some cost associated to the number of agents involved in solving a problem that we want to minimize. However, we did some further work to improve *Bounded Counsel* policy resulting in an increase of accuracy that achieves that of the Committee with a minimum number of agents involved. Although we will not pursue this here, the *proactive learning* approach explained in [7] uses induction in every agent to learn a decision tree of *voting situations*; the individually induced decision tree is used by the agent to decide whether or not to ask counsel to a new agent.

## 4 Bartering Collaboration Strategies

We have seen that agents perform better as a committee than working individually when they have a partial view of data. We can view an individual case base as a sample of examples from all examples seen by the whole multiagent system. However, in the experiments we have shown so far these individual samples were unbiased, i. e. the probability of any agent having an example of a particular solution class was equal for all agents. Nonetheless, there may be situations where the examples seen by each agent can be skewed due to external factors, and this may result in agents having a *biased* case base: i.e. having a sample of examples where instances of some class are more (or less) frequent than they are in reality.

This bias implies that individual agents have a less representative sample of the whole set of examples seen by a *MAC*. Experimental studies showed that the committee collaboration strategy decreased accuracy when the agents have biased case bases compared to the situation where their case bases are unbiased. In the following section we will formally define the notion of case base bias and show a collaboration strategy based on bartering cases that can improve the performance of a *MAC* when individual agents implement decision policies whose goal is to diminish their individual case base bias.

### 4.1 Individual Case Base Bias

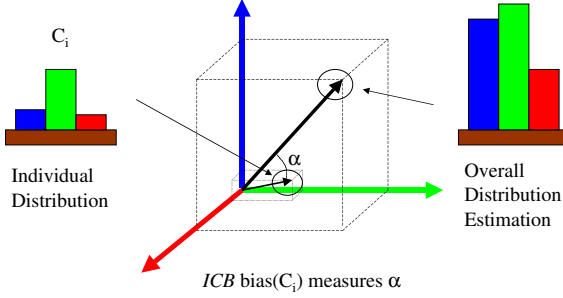
Let be  $d_i = \{d_i^1, \dots, d_i^K\}$  the individual distribution of cases for an agent  $A_i$ , where  $d_i^j$  is the number of cases with solution  $S_j \in K$  in the the case base of  $A_i$ . Now, we can estimate the overall distribution of cases  $D = \{D^1, \dots, D^K\}$  where  $D^i$  is the estimated probability of the class  $S_i$ ,  $D^j = \sum_{i=1}^n d_i^j / \sum_{i=1}^n \sum_{l=1}^K d_i^l$ .

To measure how far is the case base  $C_i$  of a given agent  $A_i$  of being a representative sample of the overall distribution we will define the *Individual Case*

*Base (ICB) bias*, as the square distance between the distribution of cases  $D$  and the (normalized) individual distribution of cases obtained from  $d_i$ :

$$ICB(C_i) = \sum_{l=1}^K \left( D^l - \frac{d_i^l}{\sum_{j=1}^K d_i^j} \right)^2$$

Figure 1 shows the cosine distance between an individual distribution and the overall distribution. The square distance is simply the distance among the normalized vectors shown in Fig. 1



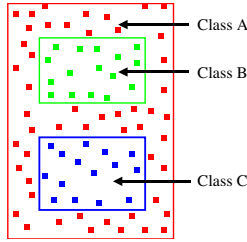
**Fig. 1.** Individual case base bias.

## 4.2 Case Bartering Mechanism

To reach an agreement for bartering between two agents, there must be an offering agent  $A_i$  that sends an offer to another agent  $A_j$ . Then  $A_j$  has to evaluate whether the offer of interchanging cases with  $A_i$  is interesting, and accept or reject the offer. If the offer is accepted, we say that  $A_i$  and  $A_j$  have reached a bartering agreement, and they will interchange the cases in the offer.

Formally an offer is a tuple  $o = \langle A_i, A_j, S_{k_1}, S_{k_2} \rangle$  where  $A_i$  is the offering agent,  $A_j$  is the receiver of the offer, and  $S_{k_1}$  and  $S_{k_2}$  are two solution classes, meaning that the agent  $A_i$  will send one of its cases (or a copy of it) with solution  $S_{k_2}$  and  $A_j$  will send one of its cases (or a copy of it) with solution  $S_{k_1}$ .

The interaction protocol in bartering is explained in [8] but essentially provides an agreed-upon pattern for offering, accepting, and performing barter actions. An agent both generates new bartering offers and assesses bartering offers received from other agents. Received bartering offers are accepted if the result of the interchange diminishes the agent's ICB. Similarly, an agent generates new bartering offers that if accepted will diminish the agent's ICB—notice, however that this effect occurs only when the corresponding agent also accepts the offer, which implies the ICB value of that agent will also diminish.



**Fig. 2.** Artificial problem used to visualize the effects of Case Bartering.

In the experiments we performed, the bartering ends when no participating agent is willing to generate any further offer, and the final state of the multiagent system is one where:

- all the individual agents have diminished their respective ICB bias values, and
- the accuracy of the committee has increased to proficient levels (as high as the levels shown in §3).

The conclusion of these experiments show that the individual decision making (based on the bias estimate) leads to an overall performance increment (the committee accuracy). Moreover, it shows that the ICB measure is a good estimate of the problems involved with the data, since “solving” the bias problem (diminishing the case base bias) has the result of solving the performance problem (the accuracy levels are restored to the higher levels we expected).

In order to have an insight of the effect of bartering in the agent’s case bases, we have designed a small classification problem for which agent’s case bases can be visualized. The artificial problem is shown in Figure 2. Each instance of the artificial problem has only two real attributes, that correspond to the  $x$  and  $y$  coordinates in the two dimensional space shown, and can belong to one of three classes (A, B or C). The goal is to guess the right class of a new point given its coordinates.

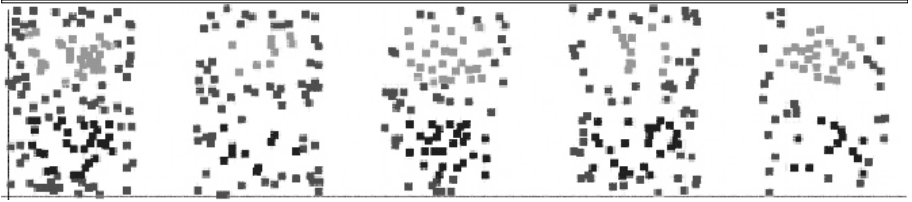
Figure 3 shows the initial cases bases of five agents for the artificial problem. Notice that the case bases given to the agents are highly biased. For instance, the first agent (leftmost) has almost no cases of the class B in its case base, and the second agent has almost only cases of class A. With a high probability, the first agent will predict class A for most of the problems for which the right solution is class B. Therefore, the classification accuracy of this agent will be very low.

Finally, to see the effect of bartering, Figure 4 shows the case bases for the same agents as Figure 3 but after applying the Case Bartering process. Notice in Fig. 4 that all the agents have obtained cases of the classes for which they had few cases before applying Case Bartering. For instance, we can see how the first agent (leftmost) has obtained a lot of cases of class B, by losing some of





**Fig. 3.** Artificial problem case bases for 5 agents before applying Case Bartering.



**Fig. 4.** Effect of the Case Bartering process in the artificial problem case bases of 5 agents.

its cases of class A. The second agent has also obtained some cases of classes B and C in exchange of losing some cases of class A.

Summarizing, each agent has obtained an individual case base that is more representative of the real problem than before applying the Case Bartering process while following an individual, self-interested decision making process.

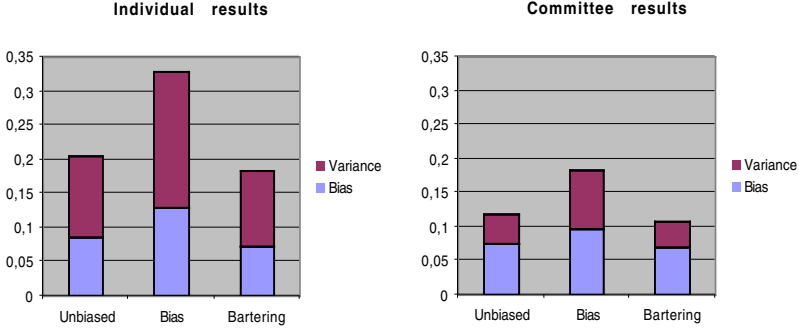
## 5 The Dimensions of Multiagent Learning

### 5.1 Bias Plus Variance Analysis

Bias plus Variance decomposition of the error [6] is a useful tool to provide an insight of learning methods. Bias plus variance analysis breaks the expected error as the sum of three non-negative values:

- Intrinsic target noise: this is the expected error of the Bayes optimal classifier (lower bound on the expected error of any classifier).
- Squared bias: measures how closely the learning algorithm's prediction matches the target (averaged over all possible training sets of a given size).
- Variance: this is the variance of the algorithm's prediction for the different training sets of a given size.

Since the first value (noise) can not be measured, the bias plus variance decomposition estimates the values of squared bias and variance. In order to estimate these values we are using the model presented in [6]. Figure 5 shows the bias plus variance decomposition of the error for a system composed of 5 agents using Nearest Neighbor. The left hand side of Figure 5 shows the bias



**Fig. 5.** Bias plus variance decomposition of the classification error for a system with 5 agents both solving problems individually and using the Committee collaboration policy.

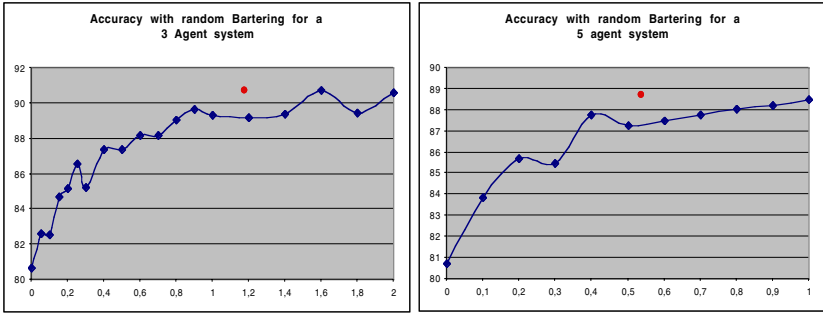
plus variance decomposition of the error when the agents solve the problems individually, and the right hand side shows the decomposition when agents use the committee collaboration policy to solve problems. Three different scenarios are presented for each one: *unbiased*, representing a situation where the agents have unbiased case bases; *biased*, representing a situation where the agents have biased case bases; *bartering*, where the agents have biased case bases and they use case bartering.

Comparing the Committee collaboration policy with the individual solution of problems, we see that the error reduction obtained with the Committee is only due to a reduction in the variance component. This result is expected since a general result of machine learning tells that we can reduce the classification error of any classifier by averaging the prediction of several classifiers when they make uncorrelated errors due to a reduction in the variance term [4].

Comparing the *unbiased* and the *biased* scenarios, we can see that the effect of the ICB bias in the classification error is reflected in both bias and variance components. The variance is the one that suffers a greater increase, but bias is also increased.

If the agents apply case bartering they can greatly reduce both components of error—as we can see comparing the *biased* and the *bartering* scenarios. Comparing the *bartering* scenario with the *unbiased* scenario, we can also see that case bartering can make agents in the *biased* scenario to achieve greater accuracies than agents in the *unbiased* scenario. Looking with more detail, we see that in the *bartering* scenario the bias term is slightly smaller than the bias term in the *unbiased* scenario. This is due to the increased size of individual case bases<sup>1</sup> because (as noted in [11]) when the individual training sets are smaller the bias

<sup>1</sup> Bartering here is realized with copies of cases, and the result is an increment on the total number of cases in the case bases of the agents. The difference between bartering with or without copy is analyzed in § 5.4



**Fig. 6.** Accuracy achieved by random bartering among 3 agents and 5 agents.

tends to increase. The variance term is also slightly smaller in the *bartering* scenario than in the *unbiased* scenario.

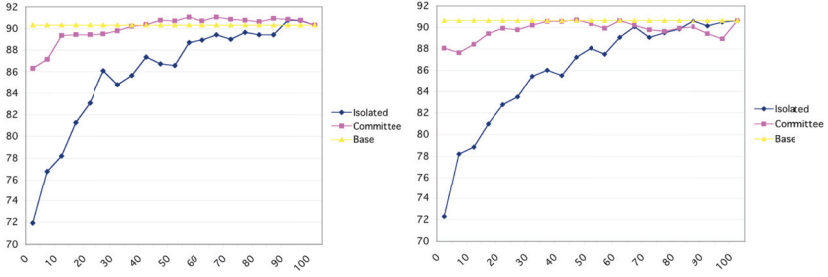
Sumarizing, the Committee collaboration policy is able to reduce the variance component of the error. Case Bartering can make a system with biased case bases to achieve grater accuracies than a system with unbiased case bases because of two reasons: 1) as the ICB bias is reduced, the accuracy of a system with unbiased case bases is recovered, and 2) as the size of individual case bases is slightly increased, the bias term of error is reduced and thus the accuracy can be greater than in the *unbiased* scenario.

## 5.2 The Effect of Individual Policies

One dimension that is interesting to assess is the effect of a specific individual decision policy inside a given collaboration strategy. In this section we shall examine the effect of the policy of diminishing ICB inside the bartering collaboration strategy.

For this purpose, we have set up an experiment to assess the difference between using the ICB policy and using a “base” (uninformed) decision policy, both with the same initial average ICB value. In the “base” experiments, the individual agents just barter cases randomly: every agent randomly chooses a percentage  $\alpha$  of cases in her case base and sends each one to another agent (also chosen at random). In this experiments,  $\alpha = 0.15$  means every agents selects at random 15% of the cases in her case base, and randomly sends each one to another agent,  $\alpha = 1$  means the agent sends all of her cases (one to each agent), and  $\alpha = 2$  means the agent sends all of her cases twice.

Figure 6 shows the accuracy of the Committee for different  $\alpha$  values on two MAC systems with 3 and 5 agents. First, notice that random bartering improves the accuracy—and the more cases are bartered (the greater the  $\alpha$ ) the higher is the accuracy for the Committee. This experiment give us the baseline utility of bartering cases in the biased scenario. However, the second thing to notice is that does not increase the accuracy as much as bartering with the ICB policy.



**Fig. 7.** Accuracy achieved by Committee using Nearest Neighbor and LID for values of  $R$  (redundancy) from 0% to 100%.

Figure 6 shows that for the same quantity of bartered cases the accuracy of the Committee is higher with the ICB policy. Moreover, notice that even when the random bartering keeps exchanging more cases (increasing  $\alpha$ ) it takes a great quantity to approach the accuracy of the ICB policy. The conclusion, thus, is that the ICB policy is capable of selecting the cases that are useful to barter among agents.

The process of random bartering introduces a lot of *redundancy* in the multi-agent system data (a great number of repeated cases in individual case bases). This is the dimension we analyze in the next section.

### 5.3 Redundancy

When we described the experiments in the Committee collaboration framework an assumption we made was that each case in our experimental dataset was adjudicated to one particular agent case base. In other words, there was no copy of any case, so redundancy in the dataset was zero. The reason we performed the experiments on the Committee under the no redundancy assumption is simply that this is the worst individual scenario (since the individual agent accuracy is lower with smaller case bases), and see how much the committee collaboration strategy could improve from there.

Let us define the *redundancy*  $R$  of a  $\mathcal{MAC}$  system as follows:

$$R = \frac{(\sum_{i=1}^n |C_i|) - M}{(n - 1)M} \cdot 100$$

where  $|C_i|$  is the number of cases in agent's  $A_i$  case base,  $n$  is the number of agents in the  $\mathcal{MAC}$  system, and  $M$  is the total number of cases. Redundancy is zero when there is no duplicate of a case, and  $R = 100$  when every agent has all ( $M$ ) cases.

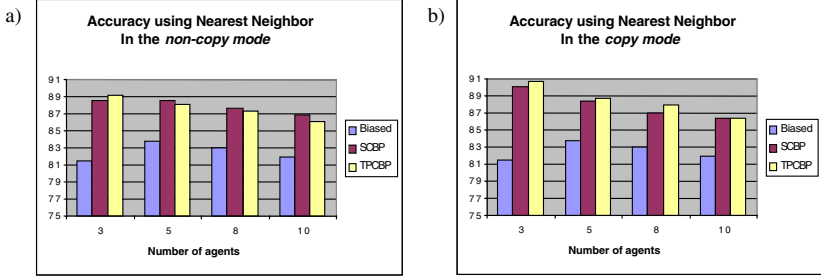
To analyze the effect of redundancy on a  $\mathcal{MAC}$  system we perform a suite of experiments shown in Fig. 7 with agents using Nearest Neighbor and LID as

CBR techniques. The experiments set up a Committee with a certain value of  $\mathbf{R}$  in the individual case bases. We show in Fig. 7 the accuracy of the Committee for different  $\mathbf{R}$  values, and we also plot there the individual (average) accuracy for the same  $\mathbf{R}$  values. The accuracy plot named “Base” in Fig. 7 is that of a single agent having all cases (i.e. a single-agent scenario). We notice that as redundancy increases the accuracy of the individual agent, as expected, grows until reaching the “Base” accuracy. Moreover, the Committee accuracy grows faster as the redundancy increases, and it reaches or even exceeds the “Base” accuracy; this fact (the Committee outperforming a single agent with all the data) is due to the “ensemble effect” of multiple model learning [5] (we discuss this further on § 6). The ensemble effect states that classifiers with uncorrelated error perform better than any one of the individual classifiers. The ensemble effect, in terms of bias plus variance, reduces the variance: that’s why Committee accuracy is higher than individual accuracy. On the other hand, individual accuracy increases with redundancy because bias is reduced. The combined effect of reducing bias and variance boosts the Committee accuracy to reach (even exceed) the “Base” accuracy (for  $\mathbf{R}$  between 50 and 75). When redundancy is very high (for  $\mathbf{R}$  higher than 90) the individual agents are so similar in the content of their case bases that to Committee strategy cannot reduce much variance, and the accuracy drops to reach the “Base” accuracy (a Committee of agents having all cases is identical to the “Base” scenario with a single agent having all cases).

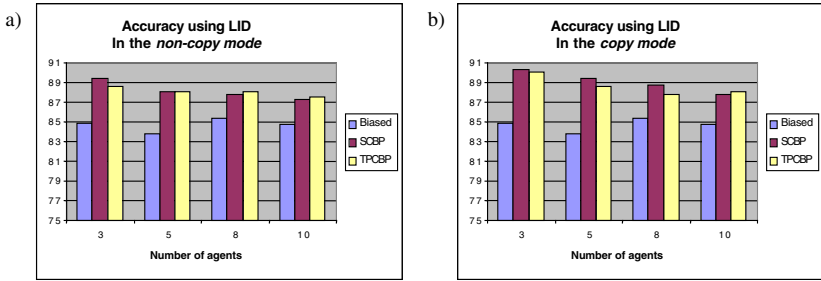
#### 5.4 Redundancy and Bartering

Redundancy also plays a role during bartering. Usually in bartering one is exchanged for the other, but since cases are simply information the barter action may involve an actual exchange of original cases or an exchange of copies of cases. Let us define *copy mode* bartering as the exchange of case copies (where bartering agents end up with both cases) and *non-copy mode* bartering as the exchange of original cases (where each bartering agent deletes the offering case and adds the receiving case). The *non-copy mode* clearly maintains the  $\mathcal{MAC}$  system redundancy  $\mathbf{R}$  while the *copy mode* increases  $\mathbf{R}$ . We performed bartering experiments both in the *copy* and *non-copy modes* and Figures 8 and 9 show the results with agents using the CBR techniques of Nearest Neighbor and LID, respectively.

Comparing now the two modes, we see that in the *non-copy mode* the  $\mathcal{MAC}$  obtains lower accuracies than in the *copy mode*. But, on the other hand, in the *non-copy mode*, the average number of cases per agent does not increase and in the *copy mode* the size of the individual case bases grows. Therefore, we can say that in the *copy mode* (when the agents send copies of the cases without forgetting them) the agents obtain greater accuracies, but at the cost of increasing the individual case base sizes. In other words, they improve the accuracy allowing case redundancy in the contents of individual case bases (a case may be contained in more than one individual case base), while in the *non-copy mode* the agents only reallocate the cases but allowing only a single copy of each case in the system.



**Fig. 8.** Accuracy in bartering using Nearest Neighbor when copying cases is allowed and disallowed.



**Fig. 9.** Accuracy in bartering using LID when copying cases is allowed and disallowed.

In terms of bias plus variance, we can see that the *copy mode* helps the individual agents to improve accuracy (since they have more cases) by decreasing the bias. This individual accuracy increment is responsible for the slight increase in accuracy of the *copy mode* versus the *non-copy mode*. Notice that the danger here for the Committee is that the “ensemble effect” could be reduced (since increasing redundancy increases error correlation among classifiers). Since bartering provides a strategy focused by the ICB policy to exchange just the cases that are most needed the redundancy increases moderately and the global effect is still positive.

## 6 Related Work

Several areas are related to our work: multiple model learning (where the final solution for a problem is obtained through the aggregation of solutions of individual predictors), case base competence assessment, and negotiation protocols. Here we will briefly describe some relevant work in these areas that is close to us.

A general result on multiple model learning [5] demonstrated that if uncorrelated classifiers with error rate lower than 0.5 are combined then the resulting

error rate must be lower than the one made by the individual classifiers. The BEM (*Basic Ensemble Method*) is presented in [9] as a basic way to combine continuous estimators, and since then many other methods have been proposed: *Bagging* [2] or *Boosting* [3] are some examples. However, all these methods do not deal with the issue of “partitioned examples” among different classifiers as we do—they rely on aggregating results from multiple classifiers that have access to *all* data. Their goal is to use a multiplicity of classifiers to increase accuracy of existing classification methods. Our intention is to combine the decisions of autonomous classifiers (each one corresponding to one agent), and to see how they can cooperate to achieve a better behavior than when they work alone. A more similar approach is the one proposed in [15], where a MAS is proposed for pattern recognition. Each autonomous agent being a specialist recognizing only a subset of all the patterns, and where the predictions were then combined dynamically.

Learning from biased datasets is a well known problem, and many solutions have been proposed. Vucetic and Obradovic [14] propose a method based on a bootstrap algorithm to estimate class probabilities in order to improve the classification accuracy. However, their method does not fit our needs, because they need the entire testset available for the agents before start solving any problem in order to make the class probabilities estimation.

Related work is that of case base competence assessment. We use a very simple measure comparing individual with global distribution of cases; we do not try to assess the areas of competence of (individual) case bases - as proposed by Smyth and McKenna [13]. This work focuses on finding groups of cases that are competent.

In [12] Schwartz and Kraus discuss negotiation protocols for data allocation. They propose two protocols, the sequential protocol, and the simultaneous protocol. These two protocols can be compared respectively to our *Token- Passing Case Bartering Protocol* and *Simultaneous Case Bartering Protocol*, because in their simultaneous protocol, the agents have to make offers for allocating some data item without knowing the other’s offers, and in the sequential protocol, the agents make offers in order, and each one knows which were the offers of the previous ones.

## 7 Conclusions and Future Work

We have presented a framework for Cooperative Case-Based Reasoning in multiagent systems, where agents use a market mechanism (bartering) to improve the performance both of individuals and of the whole multiagent system. The agent autonomy is maintained, because each agent is free to take part in the collaboration processes or not. For instance, in the bartering process, if an agent does not want to take part, he just has to do nothing, and when the other agents notice that there is one agent not following the protocol they will ignore it during the remaining iterations of the bartering process.

In this work we have shown a problem arising when data is distributed over a collection of agents, namely that each agent may have a skewed view of the world (the individual bias). Comparing empirical results in classification tasks we saw that both the individual and the overall performance decreases when bias increases. The process of bartering shows that the problems derived from distributed data over a collection of agents can be solved using a market-oriented approach. Each agent engages in a barter only when it makes sense for its individual purposes but the outcome is an improvement of the individual and overall performance.

The naive way to solve the ICB bias problem could be to centralize all data in one location or adopt a completely cooperative multiagent approach where each agent sends its cases to other agents and they retain what they want (a “gift economy”). However, these approaches have some problems; for instance, having all the cases in a single case base may not be practical due to efficiency problems. Another problem of the centralized approach is that the agents belong to organizations that consider their case bases as assets, they are not willing to donate their cases to a centralized case base. Case Bartering tries to interchange cases only to the amount that is necessary and not more, to keep the redundancy not increasing very much.

As a general conclusion, we have seen that there are avenues to pursue the goal of learning systems, in the form of multiagent systems, where the training data need not be centralized in one agent nor duplicated in all agents. New non-centralized processes can be designed that are able to correct problems in that distributed allocation of training data, for instance bartering. We have seen that the “ensemble effect” of multi-model learning also takes place in the multiagent setting, even in the situation where there is no redundancy.

Finally, we have focused on lazy learning techniques (CBR) because it seemed easier to be adapted to a distributed, multiagent setting; however, the same ideas and techniques should be able to work for multiagent systems that learn using eager techniques like induction. We plan to investigate inductive multiagent learning in the near future, starting with classification tasks and decision tree techniques.

## References

1. E. Armengol and E. Plaza. Lazy induction of descriptions for relational case-based learning. In *12th European Conference on Machine Learning*, 2001.
2. Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.
3. Yoav Freund and Robert E. Schapire. Experiments with a new boosting algorithm. In *Proc. 13th ICML*, pages 148–146. Morgan Kaufmann, 1996.
4. Jerome H. Friedman. On bias, variance,  $0/1$  – loss, and the curse-of-dimensionality. *Data Mining and Knowledge Discovery*, 1(1):55–77, 1997.
5. L. K. Hansen and P. Salamon. Neural networks ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (12):993–1001, 1990.
6. Ron Kohavi and David H. Wolpert. Bias plus variance decomposition for zero-one loss functions. In Lorenza Saitta, editor, *Machine Learning: Proceedings of the Thirteenth International Conference*, pages 275–283. Morgan Kaufmann, 1996.



7. S. Ontañón and E. Plaza. Learning when to collaborate among learning agents. In *12th European Conference on Machine Learning*, 2001.
8. S. Ontañón and E. Plaza. A bartering approach to improve multiagent learning. In *1st International Joint Conference in Autonomous Agents and Multiagent Systems*, 2002.
9. M. P. Perrone and L. N. Cooper. When networks disagree: Ensemble methods for hybrid neural networks. In *Artificial Neural Networks for Speech and Vision*. Chapman-Hall, 1993.
10. E. Plaza and Ontañón S. Ensemble case-based reasoning. *Lecture Notes in Artificial Intelligence*, 2080:437–451, 2001.
11. Richard A. Olshen and L. Gordon. Almost sure consistent nonparametric regression from recursive partitioning schemes. *Multivariate Analysis*, 15:147–163, 1984.
12. R. Schwartz and S. Kraus. Bidding mechanisms for data allocation in multi-agent environments. In *Agent Theories, Architectures, and Languages*, pages 61–75, 1997.
13. B. Smyth and E. McKenna. Modelling the competence of case-bases. In *EWCBR*, pages 208–220, 1998.
14. S. Vucetic and Z. Obradovic. Classification on data with biased class distribution. In *12th European Conference on Machine Learning*, 2001.
15. L. Vuurpijl and L. Schomaker. A framework for using multiple classifiers in a multiple-agent architecture. In *Third International Workshop on Handwriting Analysis and Recognition*, 1998.